

爱创课堂前端培训

CSS

第四天

班级： 爱创课堂 7 期

日期： 2017 年 4 月 18 日

目录

HTML.....	错误！未定义书签。
目录.....	1

一、复习

1、盒模型的扩展

- 宽度剩余法

```
1 padding-left:20px;  
2 padding-right:0;
```

- height:通常盒子高度不限定，通过内容撑高我们盒子。
- 居中：

单行文本的垂直居中：

```
1 height:20px;  
2 line-height:20px;
```

多行文本垂直居中：

```
1 padding:20px 0;
```

文本水平居中：

```
1 text-align:center;
```

盒子水平居中：

```
1 width:500px;
2 margin:0 auto;
```

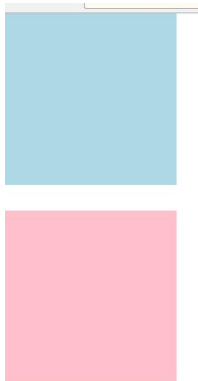
盒子垂直居中：

```
1 padding:20px 0;
```

- margin:

margin 有垂直方向的塌陷：两个上下排列盒子，上盒子有一个下 margin，下盒子有一个上 margin。他们之间距离是大 margin 的数值（也就是说小 margin 会大 margin 里）

```
1 .top{
2     width: 200px;
3     height: 200px;
4     background-color: lightblue;
5     margin-bottom:30px;
6 }
7 .bottom{
8     width: 200px;
9     height: 200px;
10    background-color: pink;
11    margin-top:30px;
12 }
```



结果：距离是 30px;

不能用儿子去踹父亲：父子盒模型中，不能用儿子的 margin-top 去撑开上间距，而是用父盒子的 padding-top 挤。

```
1 padding-top:20px;
```

- 书写网页时第一步就是清除默认样式。
- 父子盒模型：子盒子的最大占有区域不能超过父盒子的内容宽高。

2、标准文档流

网页是标准文档流，书写只能是从上到下，从左到右。

文本类内容有空白折叠现象；

文本类内容高矮不齐，底部对齐；

文本类内容超过盒子宽度会自动换行；

块级元素：

独占一行；

可以设置宽高;

不设置宽, 是父盒子宽度的 100%;

```
1 div,h1-h6,p,ul,ol,li
```

行内元素:

行内元素会显示在一行,

不能设置宽高;

宽度是内容多少;

```
1 span,a ,input,b,i
```

块级元素和行内元素的相互转换: `display:(显示模式)`

```
1 display:block (转为块) 常用
```

```
2 display:inline;(转为行内)不常用
```

脱离标准流: 浮动, 绝对定位, 固定定位。

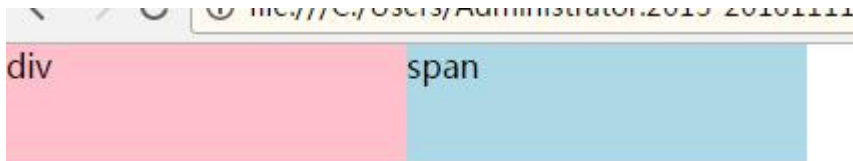
```
1
```

二、浮动的性质

1、浮动的元素脱离标准流

浮动的元素脱离标准流, 不再区分块级元素和行内元素。能够让浮动的元素并排在一行显示, 还可以设置宽高。

```
1 div{
2     float:left;
3     width:200px;
4     height:60px;
5     background-color: pink;
6 }
7 span{
8     float: left;
9     width:200px;
10    height:60px;
11    background-color: lightblue;
12 }
```



2、浮动的元素没有 *margin* 塌陷

```
1 .box .son1{
2     width: 200px;
3     height: 200px;
4     background-color: lightblue;
5     margin-bottom:50px;
6     float: left;
7 }
8 .box .son2{
9     float: left;
10    width: 200px;
11    height: 200px;
```

```

12     background-color: pink;
13     margin-top: 80px;
14 }

```



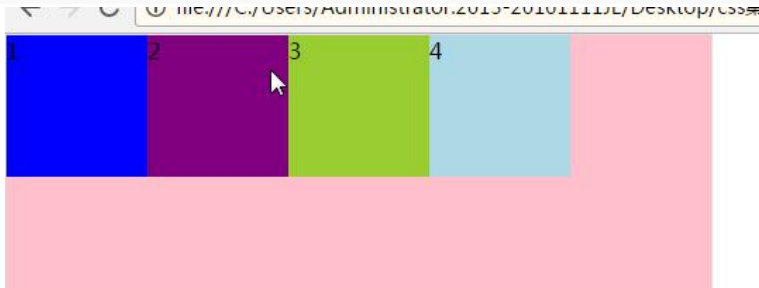
3、浮动的元素依次贴边

浮动的元素依次贴边，贴边有方向之分。

以左浮动为例：

- 贴边顺序：父盒子内侧 ← 子盒子 1 ← 子盒子 2 ← 子盒子 3 ← 子盒子 4....

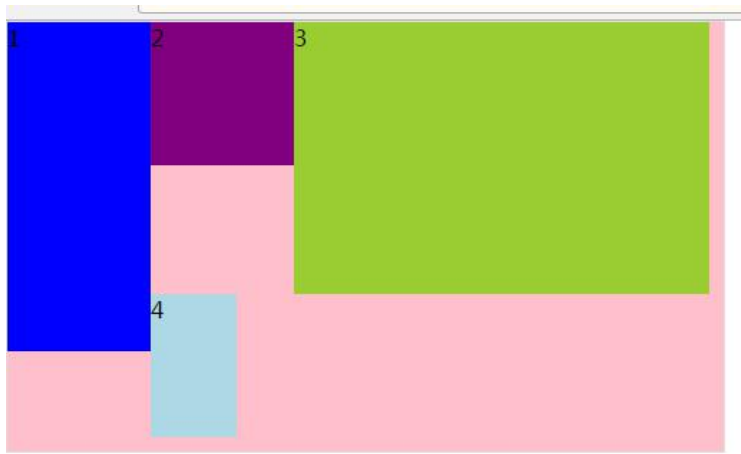
1



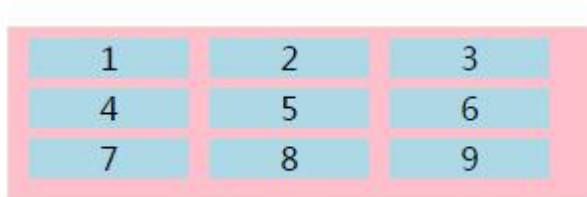
- 如果前一个盒子剩余的宽度不够，会再前一个贴边。



- 浮动的元素不会钻盒子：

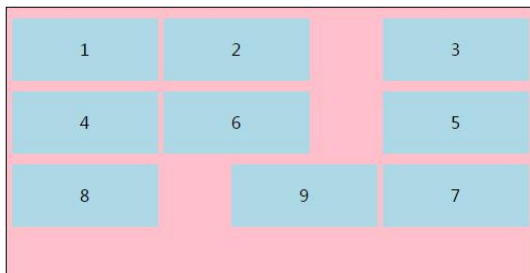


- 小例子：表格项



- 双向贴边

1



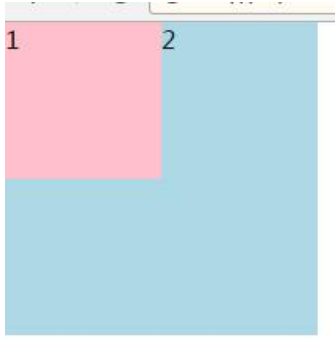
4、浮动的元素让出标准流的位置

俩个同级的盒子，前面的盒子左浮动，后面的盒子不浮动。浮动的元素会让出标准流的位置。

```

1      .box1{
2          float:left;
3          width:100px;
4          height:100px;
5          background-color: pink;
6      }
7      .box2{
8          width: 200px;
9          height: 200px;
10         background-color: lightblue;
11     }

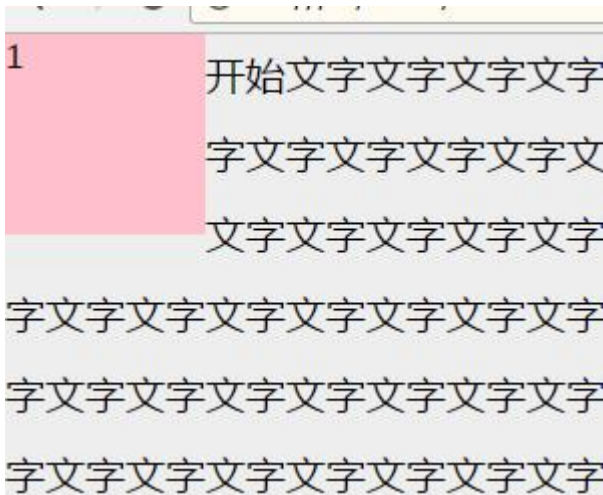
```



浮动的元素会让出标准流的位置，视觉上有一个压盖效果。（实际工作中我们不会用浮动做压盖效果，用定位做压盖效果。）

5、字围效果

俩个同级的盒子，前面的盒子左浮动，后面的盒子不浮动。不浮动的盒子里面的文字不会压盖住，会在浮动的盒子周围显示。



三、清除浮动

1、浮动存在的问题

- ①浮动的元素不能撑高父盒子。
- ②浮动的元素会影响后面浮动的元素。

2、清除浮动的办法

1) 给父盒子加高度

```
1 .box1,.box2{
2     width:700px;
3     border:5px solid #ddd;
4     margin-bottom:10px;
5     height:200px;
6 }
```



解决:

对后面浮动元素的影响; margin 有距离。

没解决:

高度自适应。

2) 加 clear 属性

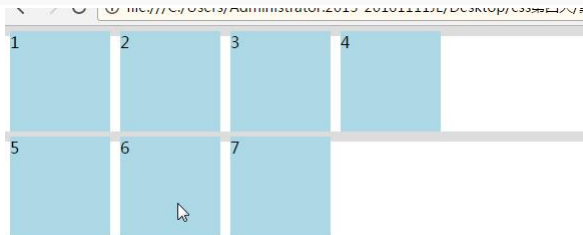
clear:清除;

属性值: left (清除左浮动元素的影响)

right (清除右浮动元素的影响)

both(清除左右浮动元素的影响)。

```
1 .box2{
2   clear:both;
3 }
```



解决:

浮动元素对后面浮动元素的影响。

没解决:

高度自适应;

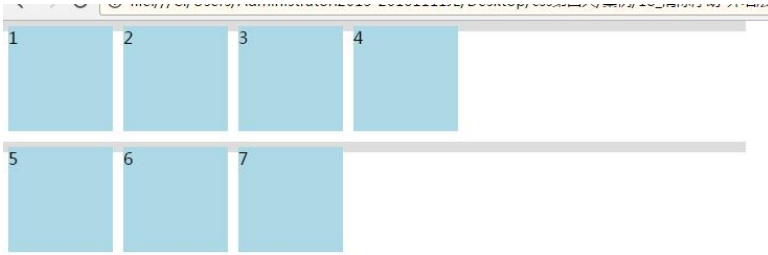
margin 距离

3) 隔墙法

● 外墙法:

给俩个大盒子之间加一堵清除了浮动, 有高度的墙。

```
1 <div class="box1">
2   <p>1</p>
3   <p>2</p>
4   <p>3</p>
5   <p>4</p>
6 </div>
7 <div class="cl h10"></div>
8 <div class="box2">
9   <p>5</p>
10  <p>6</p>
11  <p>7</p>
12 </div>
```



解决:

浮动元素对后面浮动元素的影响;

视觉有了距离;

没解决:

高度自适应

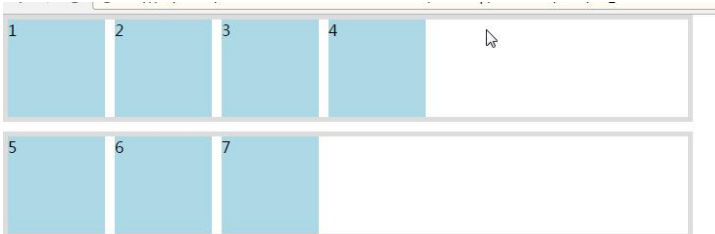
● 内墙法:

在两个大盒子的最底部加一堵清除了浮动的墙。

```

1    <div class="box1">
2        <p>1</p>
3        <p>2</p>
4        <p>3</p>
5        <p>4</p>
6        <div class="cl"></div>
7    </div>
8    <div class="box2">
9        <p>5</p>
10       <p>6</p>
11       <p>7</p>
12       <div class="cl"></div>
13    </div>

```



解决: 都解决了。

虽然内墙法解决了我们所有问题, 但是增加了很多无意义的标签。影响代码加载速度以及可读性。

4) *overflow* 属性

overflow: 溢出处理方法。

属性值:

auto: 溢出滚动;

hidden: 溢出隐藏;

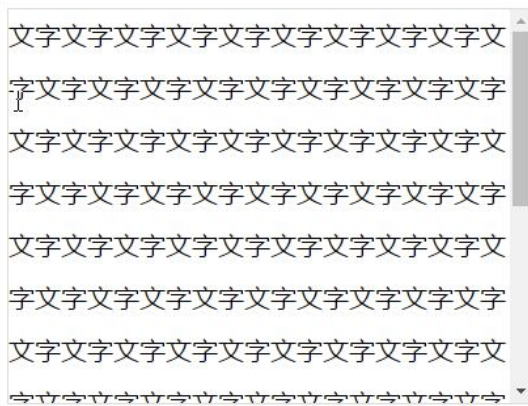
1

不设置任何 *overflow* 属性值:

[illegible]

```
1 overflow:auto;
```

当文字超出盒子高度时，会自动显示滚动条，拉动滚动条可以查看全部文字。



```
1  overflow:hidden;
```

当文字超出盒子高度时，文字自动隐藏。（隐藏的是 **border 边框外** 的文字。）

注意：溢出隐藏，对于下 padding 的部分并不能隐藏，隐藏 border 边框外面的部分。



清除浮动：给父盒子加 `overflow:hidden`;

```
1      .box1,.box2{
2          width:700px;
3          border:5px solid #ddd;
4          margin-bottom:10px;
5          overflow: hidden;
```



解决：（实际工作中真正使用的。）

- 高度自适应；
- 浮动元素对后面浮动元素的影响；
- margin 有距离；

overflow: hidden 作用

- ①强制我们父盒子去检测里面的子盒子的高度，让父盒子的高度等于子盒子的高度。（以后只要看到让盒子高度自适应，就加 overflow:hidden）
 - ②强制我们父盒子去检测他里面浮动的子盒子，管住他里面浮动的元素不让他去影响别人。
- （**overflow: hidden**是我们解决浮动存在的问题的最好也是最简单的办法）

实际工作中：

- 我们使用 **overflow: hidden**解决浮动存在的问题；
- 可以在几个大的板块之间加一堵墙；

小案例：选项卡（tab 栏）



1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1
1
1
1
1
1
1